AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions of claims in the application:

Listing of Claims:

1. (Original) A user interface automation system comprising:

an input component that receives a request; and,

a navigation component that receives the request from the input component and facilitates simulated user interface associated with an automation component based, at least in part, upon information stored in a map information store and information stored in a command information store.

2. (Original) The system of claim 1, wherein the automation component is a wizard.

3. (Original) The system of claim 1, wherein the map information store comprises a text-based file.

4. (Currently amended) The system of claim 1, wherein the ~~configuration~~ command information store comprises a text-based file.

5. (Original) The system of claim 1, wherein the navigation component further facilitates simulated user interface based, at least in part, upon information stored in a global information store.

6. (Original) The system of claim 5, the navigation component employing information stored in the global information store when a global variable is encountered in the command information store.

7. (Original) The system of claim 1, wherein at least one of the map information store and the configuration information store comprise at least one alias name.

8. (Original) The system of claim 1, wherein the navigation component further stores error information in a log information store.

9. (Original) The system of claim 1, wherein the navigation component further stores information associated with the request in a log information store.

10. (Original) The system of claim 9, wherein the navigation component iterates through information stored in the command information store, performs the indicated operation and stores information associated with the indicated operation in the log information store.

11. (Original) The system of claim 9, wherein the navigation component stores error information in the log information store.

12. (Original) The system of claim 1, wherein the input component performs input validation upon the request and provides error information if the request is invalid.

13. (Original) The system of claim 12, wherein a graphical message is displayed to a user of the system, the graphical message being based, at least in part, upon the error information from the input component.

14. (Original) The system of claim 1, wherein the input component receives a command line invocation.

15. (Original) The system of claim 1, the map information store comprising a section name and a page identifier.

16. (Original) The system of claim 15, the page identifier comprising a label for a control, the page identifier further uniquely identifying a particular page.

17. (Original) The system of claim 15, the page identifier comprising a control type.

18. (Original) The system of claim 17, wherein the control type is at least one of button, combo, list, scroll, static, radio and check.

19. (Original) The system of claim 1, wherein information stored in the command information store can be modified by at least one of a front-end user interface application, scripting, a batch file and a text editor.

20. (Original) The system of claim 1, the command information store comprising a section name, the section name corresponding to information stored in the map information store, the command information store further comprising an action.

21. (Original) The system of claim 1, the command information store storing information associated with at least one of a function key and a control key simulation.

22. (Original) A method of automating user interface comprising:
receiving mapping information from a map information store;
receiving command information from a command information store;
performing simulated user interface based, at least in part, upon information stored in the map information store and the command information store.

23. (Original) The method of claim 22, further comprising:
storing information in a log information store, if an error is detected performing the simulated user interface.

24. (Original) A computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 22.

25. (Original) A method of automating user interface comprising:
retrieving mapping information from a map file;
retrieving command information from a command file;
obtaining a section name from the command file;

retrieving page identification information from the map file associated with the section name;

retrieving section data for section associated with the section name from the command file; and,

performing an action associated with the retrieved section data.

26. (Original) The method of claim 25, further comprising:

storing information in a log file, if an error is detected performing the action.

27. (Original) A computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 25.

28. (Original) A user interface automation system comprising:

an input component that receives a request; and,

a navigation component that receives the request from the input component and facilitates simulated user interface associated with an automation component based, at least in part, upon information stored in a map information store and information stored in a command information store.

29. (Original) A user interface automation system comprising:

means for receiving a request; and,

means for simulating user interface associated with an automation component based, at least in part, upon information stored in a map information store and information stored in a command information store, the means for simulating receiving the request from the means for receiving.

30. (Currently amended) A tangible data packet transmitted between two or more computer components that facilitates user interface simulation, the data packet comprising:

a section name and a page identifier that uniquely identifies a particular page, the page identifier comprising a label for a control and a control type.

31. (New) The method of claim 22, further storing data, commands and executables associated with the user interface separately.

32. (New) The method of claim 31, facilitates a modular system which can be modified without recompilation of the executable.